

Redes y Sistemas Complejos (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 3

Estudio Comparativo de Métodos para Poda y Visualización de Redes.

Braulio Vargas López
DNI: 20079894C
Correo: brauliovarlop@correo.ugr.es

28 de noviembre de 2016

ÍNDICE

1. Poda y Visualización	1
1.1. Visualizaciones de las Redes	2
2. Análisis de eficiencia de las variantes del algoritmo <i>Pathfinder</i>	5

1. PODA Y VISUALIZACIÓN

En esta primera parte del guión, vamos a realizar un análisis de diez de las veinte redes disponibles, podando cada una de las redes con la versión *BinaryPathfinder* del algoritmo. Las diez redes escogidas son:

1. France-2002
2. Germany-2002
3. Japan-2002
4. Spain-1996
5. Spain-1998
6. Spain-2002
7. Spain-2004
8. United_Kingdom-2002
9. United_States-2002
10. World

La razón de coger estas redes es por el hecho de poder ver la evolución de la producción científica en España a lo largo de ocho años, y poder compararlas con otros países durante el año 2002. Además, de poder ver la producción científica a nivel mundial en la última red. A continuación, podemos ver los resultados obtenidos aplicando el algoritmo a cada uno de las redes seleccionadas:

France-2002			Germany-2002		
n=267	<i>L</i>	<i>D</i>	n=269	<i>L</i>	<i>D</i>
Red original	23986	0.675453	Red original	25395	0.704516
2	312	0.00878601	2	313	0.00868335
3	275	0.00774408	3	277	0.00768463
4	271	0.00763144	4	272	0.00754591
5	270	0.00760328	5	270	0.00749043
266	268	0.00754696	268	269	0.00746269

Tabla 1.1: Tablas para las redes de France-2002.net y Germany-2002.net

Japan-2002			Spain-1996		
n=265	<i>L</i>	<i>D</i>	n=243	<i>L</i>	<i>D</i>
Red original	21754	0.621898	Red original	5967	0.202938
2	316	0.00903373	2	394	0.0134
3	279	0.00797599	3	313	0.0106452
4	269	0.00769011	4	303	0.0103051
5	267	0.00763293	5	300	0.010203
264	267	0.00763293	242	300	0.010203

Tabla 1.2: Tablas para las redes Japan-2002.net y Spain-1996.net

Spain-1998			Spain-2002		
n=258	L	D	n=264	L	D
Red original	12971	0.391247	Red original	21807	0.628154
2	320	0.00965222	2	320	0.00921765
3	279	0.00841553	3	274	0.00789261
4	267	0.00805357	4	265	0.00763337
5	267	0.00805357	5	263	0.00757576
257	267	0.00805357	263	263	0.00757576

Tabla 1.3: Tablas para las redes *Spain-1998.net* y *Spain-2002.net*.

Spain-2004			United_Kingdom-2002		
(n=269)	L	D	(n=276)	L	D
Red original	24991	0.693309	Red original	28707	0.756443
2	332	0.00921045	2	326	0.00859025
3	280	0.00776785	3	288	0.00758893
4	272	0.00754591	4	280	0.00737813
5	271	0.00751817	5	279	0.00735178
268	270	0.00749043	275	276	0.00727273

Tabla 1.4: Tablas para las redes *Spain-2004.net* y *United_Kingdom-2002.net*.

United_States-2002			World.net		
(n=276)	L	D	(n=218)	L	D
Red original	31292	0.824559	Red original	20154	0.85207
2	314	0.00827404	2	280	0.0118378
3	287	0.00756258	3	233	0.00985076
4	279	0.00735178	4	223	0.00942798
5	277	0.00729908	5	220	0.00930115
275	275	0.00724638	217	217	0.00917431

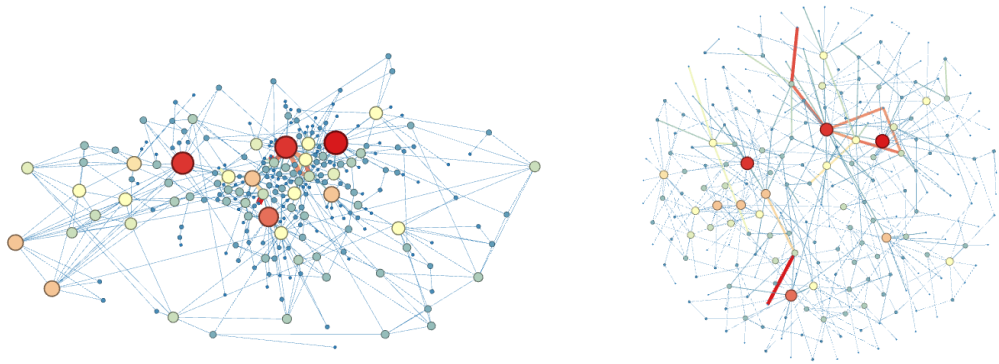
Tabla 1.5: Tabla para las redes *United_States-2002.net* y *World.net*.

Como podemos ver en las tablas anteriores, los resultados para todas las redes son parecidos. Con $n = 2$, el algoritmo poda una gran cantidad de enlaces en todas las redes, lo que hace bajar muchísimo la densidad de la red. Por ejemplo, en la [Tabla 1](#), para la red *France-2002.net* podemos ver cómo la red original tenía 23986 enlaces y una densidad $D \approx 0,7$, pasa a tener con 312 enlaces y una densidad de $D \approx 0,009$, unas 77 veces menor aproximadamente. Además de esto, en todas las redes se da el mismo suceso y es que el número de enlaces para $n = 5$ y $n = \infty$ es prácticamente igual, o hay casos en los que es igual, como se puede ver en las tablas [Tabla 1](#). Esto se debe a que el algoritmo, a partir de este punto, no puede podar más.

1.1. VISUALIZACIONES DE LAS REDES

Las redes que vamos a visualizar son las redes de *Spain-1996.net* y *Spain-2004.net*. A continuación, podremos ver las distintas visualizaciones para los algoritmos **Frucherman & Reingold** y **Kamada & Kawai**. Además de esto, servirá para ver la evolución científica en España tras 8 años.

En ambas redes, podremos ver de color más rojo y más grande los nodos que más grado tienen. De igual manera pasa con los enlaces de la red. Serán de mayor grosor y más rojos cuanto mayor sea el peso del enlace.

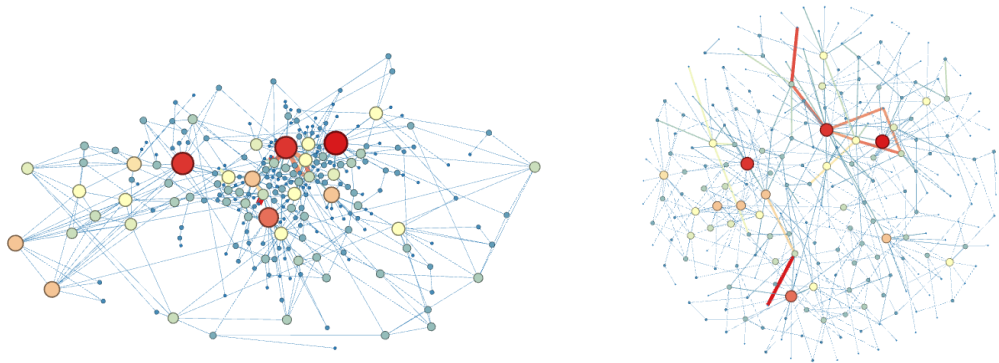


(a) Distribución de nodos con Kamada & Kawai.

(b) Distribución de nodos con Frutcherman & Reingold.

Figura 1.1: Kamada & Kawai y Frutcherman & Reingold para la red Spain-1996 con $q = 2$.

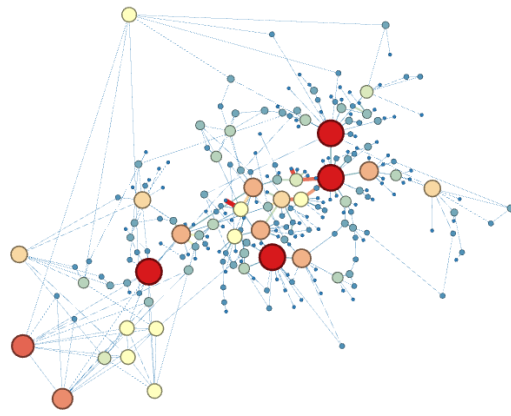
En la [Figura 2\(a\)](#) podemos ver como el algoritmo Kamada & Kawai hace una distribución “similar” a un plano de metro, donde podemos ver de forma más clara, la distribución de la producción científica del país. En los nodos centrales (si el tamaño permitiera ver bien las etiquetas), podemos ver que los nodos más importantes son la medicina y la bioquímica, lléndose hacia la izquierda ramas científicas como la ingeniería, matemática aplicada... Sin embargo, Frutcherman & Reingold ([Figura 2\(b\)](#)) realiza una distribución circular, donde es más difícil interpretar los datos, ya que todos quedan más o menos a la misma distancia, cosa que con Kamada & Kawai no pasa, ya que aleja los nodos que menos relación tienen entre sí.



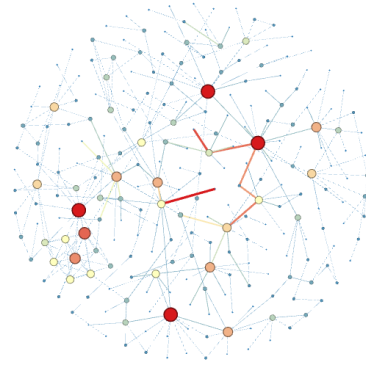
(a) Distribución de nodos con Kamada & Kawai.

(b) Distribución de nodos con Frutcherman & Reingold.

Figura 1.2: Kamada & Kawai y Frutcherman & Reingold para la red Spain-1996 con $q = 2$.

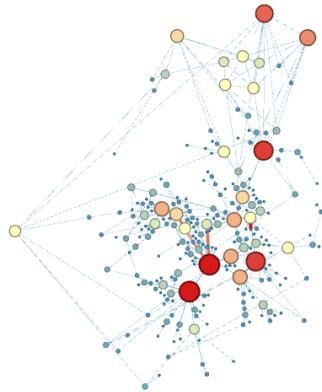


(a) Distribución de nodos con Kamada & Kawai.

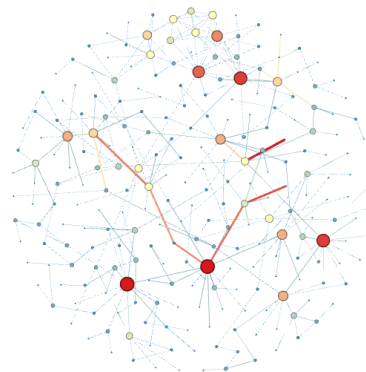


(b) Distribución de nodos con Fruchterman & Reingold.

Figura 1.3: Kamada & Kawai y Fruchterman & Reingold para la red Spain-1996 con $q = 3$.



(a) Distribución de nodos con Kamada & Kawai.



(b) Distribución de nodos con Fruchterman & Reingold.

Figura 1.4: Kamada & Kawai y Fruchterman & Reingold para la red Spain-1996 con $q = 4$.

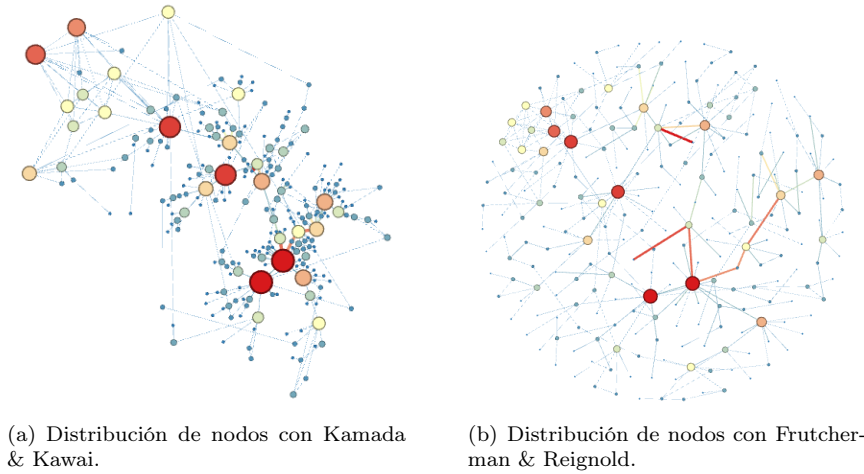


Figura 1.5: Kamada & Kawai y Frutcherman & Reingold para la red Spain-1996 con $q = n - 1$.

Como se puede ver en cada una de las imágenes podemos ver cómo, para este tipo de red, el algoritmo Kamada & Kawai realiza una mejor visualización que Frutcherman & Reingold, dejando más alejadas las materias que no tienen relación entre sí, y más cercanas las que sí tienen relación, mientras que Frutcherman & Reingold, realiza una visualización circular, al querer dejar los enlaces con la misma longitud.

2. ANÁLISIS DE EFICIENCIA DE LAS VARIANTES DEL ALGORITMO *Pathfinder*

Para analizar la eficiencia práctica de los distintos algoritmos, vamos a lanzarlos para redes aleatorias de tamaño 500, 1000, 2000, 5000 y 10000 nodos. Para ello, se ha creado un script que ejecuta los algoritmos para las distintas redes y genera una tabla en la que podemos ver los resultados de las distintas ejecuciones.

N	$ L $	$ D $	Original	Binary	Fast	MST (Teórico)	MST (Práctico)
500	1252	0.01003	37,71665	4,40057	0,29051	0,00333	0,00296
1000	4999	0.01000	617,89157	49,98381	2,22931	0,01109	0,01079
2000	20041	0.01002	> 1800	750,72829	18,83852	0,17109	0,10172
5000	124525	0.00996	> 1800	> 1800	276,64002	0,53278	0,48843
10000	498666	0.00997	> 1800	> 1800	> 1800	2,53242	2,49397

Tabla 2.1: Tabla de resultados para los distintos algoritmos.

Como se puede ver en la [Tabla 2.1](#), la versión original del algoritmo *Pathfinder*, es bastante ineficiente, en comparación con el resto, ya que para problemas de pequeño tamaño, ya requiere una gran cantidad de tiempo y un gran espacio. Esto se debe, al estar basado en programación dinámica, tienen unos ordenes de complejidad en tiempo de $\mathcal{O}(n^4)$ y una complejidad en espacio de $\mathcal{O}(n^3 - 2 \cdot n^2) \equiv \mathcal{O}(n^3)$, que aunque sea polinómica, para problemas donde n , ni siquiera es grande, con un $n = 1000$ los tiempos de ejecución son del orden de unos 10 minutos en una máquina con un procesador que funciona con *overclock* a $4,6GHz$.

La segunda versión del algoritmo *Pathfinder*, *BinaryPathfinder*, mejora estos órdenes de complejidad del algoritmo, pero aun así, para problemas de gran tamaño, sigue sufriendo y es incapaz de acabar en tiempos razonables. Esto se debe a que, a pesar de haber rebajado los órdenes de complejidad, estos siguen siendo $\mathcal{O}(n^3 \log n)$ para el tiempo, y $\mathcal{O}(4 \cdot n^2)$ para el espacio, que aunque sean menores, siguen siendo “malos” para problemas donde n es grande.

En la versión *FastPathfinder*, se vuelve a reducir tanto la complejidad en tiempo como en espacio, pasando a tener $\mathcal{O}(n^3)$ para el tiempo, y $\mathcal{O}(3 \cdot n^2)$ para el espacio. Esta reducción permite que para problemas donde n es pequeña, pase a tiempos de décimas de segundo, y nos permite ejecutar problemas donde $n = 5000$, en unos 5 minutos aproximadamente. Aun así, a pesar de esta mejora, seguimos poder sin ejecutar problemas donde la n tiene un tamaño medio en un tiempo razonable, es decir, $n > 7000$.

Estas tres versiones utilizan la programación dinámica como algoritmo base, lo que provoca estos órdenes de complejidad tanto en tiempo como en espacio. Las siguientes versiones cambian de paradigma, abandonando la programación dinámica y usando un algoritmo de la teoría clásica de grafos: **el algoritmo de Kruskal**.

El algoritmo de Kruskal es un algoritmo clásico de la teoría de grafos, muy utilizado para encontrar el árbol generador minimal de un grafo G que toma como entrada, ya que la complejidad en tiempo y espacio es muy pequeña, lo que lo hace muy eficiente. Tomando esto como base, surgieron los siguientes algoritmos.

La versión teórica de esta versión, *MST-Theoretical*, es exactamente igual a la versión *MST-Practical* pero cambia la estructura de datos utilizada en los algoritmos. La versión teórica tiene un orden de complejidad menor que la versión práctica, pero los tiempos de ejecución son mayores debido a la estructura de datos que tiene asociada. La versión teórica usa una estructura de datos de conjuntos disjuntos con una unión por rango y comprensión de caminos, mientras que la versión práctica, utiliza una lista indexada de conjuntos disjuntos, es decir, tres listas de enlaces con sus pesos (F , T y H) junto con el índice. Para más detalle, podemos consultar el artículo original [?].

Los órdenes de complejidad son $\mathcal{O}(n^2 \cdot \log n)$ para el tiempo y $\mathcal{O}(3 \cdot n^2 + n)$ para el espacio en la versión teórica del algoritmo MST, mientras que para la versión práctica tenemos $\mathcal{O}(n^3)$ para el tiempo y $\mathcal{O}(3 \cdot n^2 + n)$ para el espacio. Como podemos ver, la versión teórica tiene unos órdenes de complejidad menor que la versión práctica, pero aun así, gracias a que las estructuras de datos son más simples en la versión práctica, podemos observar en la [Tabla 2.1](#) la versión práctica es más rápida que la teórica.

A pesar de la mejora en tiempo, supone un deterioro en los órdenes de complejidad en espacio, pero es un precio a pagar para poder representar redes con un mayor número de nodos en un tiempo razonable.